# Upgrade Testing

This page describes upgrade procedures to completely test the client and server. These procedures can be used for either pre-release testing, or for confidence building in a new release.

This page is a work in progress.

## Server Testing

1. Ensure that the server produces the same configurations for clients

- Before the upgrade, generate all client configurations using the `buildall` subcommand of `bcfg2-info`. This subcommand takes a directory argument; it will generate one client configuration in each file, naming each according to the client name.

```
mgt1:~/bcfg# bcfg2-info
Filesystem check 1 of 25
...
> buildall /path/to/cf-old
Generated config for fs2.bgl.mcs.anl.gov in 1.97310400009 seconds
Generated config for fs13.bgl.mcs.anl.gov in 1.47958016396 seconds
...
```

  Take notice of any messages produced during configuration generation. These generally reflect minor issues in the configuration specification. Ideally, they should be fixed.

- Upgrade the server software
- Generate all client configurations in a second location using the new software.
  Any tracebacks reflect bugs, and should be filed in the ticketing system. Any new messages should be carefully examined.
- Compare each file in the old directory to those in the new directory using `bcfg2-admin compare -r /old/directory /new/directory

```
mgt1:~/bcfg# bcfg2-admin compare -r cf-old/ cf-new/
Entry: fs2.bgl.mcs.anl.gov.xml
Entry: fs2.bgl.mcs.anl.gov.xml good
Entry: fs13.bgl.mcs.anl.gov.xml
Entry: fs13.bgl.mcs.anl.gov.xml good
Entry: login1.bgl.mcs.anl.gov.xml
 ConfigFile /bin/whatami contents differ
 ConfigFile /bin/whatami differs (in bundle softenv)
Entry: login1.bgl.mcs.anl.gov.xml bad
```

  This can be used to compare configurations for single clients, or different clients.

2. Compare old and new group diagrams (using bcfg2-admin viz)

## Client Testing

1. Run the client in dry-run and non-dry-run mode; ensure that multiple runs produce consistent results.